

Linguagem C#

Programação Visual

Sumário

- Conceitos básicos da linguagem C# em comparação com as linguagens Java e C++.
- Conceitos avançados da linguagem C#
- **A linguagem integrada de interrogação de dados – LINQ**

LINQ

Sessão 1 – Introdução ao LINQ

LINQ básico, execução diferida e imediata.



```
public class Student
{
    public string Name { get; set; }
    public int Number { get; set; }

    public Student()
    {
    }

    public Student(string name, int number)
    {
        Name = name;
        Number = number;
    }

    public override string ToString()
    {
        return $"Name: {Name}, Number: {Number}";
    }
}
```

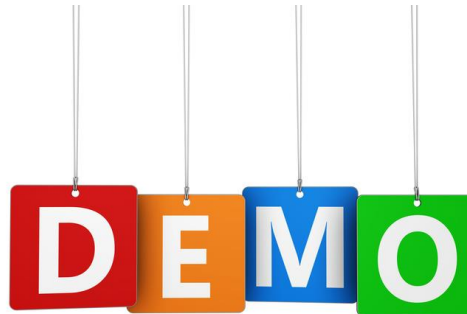


```
List<Student> stds = new List<Student> {  
    new Student {Name = "Joao Carlos", Number = 1234},  
    new Student {Name = "Jose Antunes", Number = 2345},  
    new Student {Name = "Joao Silva", Number = 3456},  
    new Student {Name = "Ana Costa", Number = 4567},  
    new Student {Name = "Ivo Tiago", Number = 5678},  
    new Student {Name = "Manuel Jose", Number = 6789},  
    new Student {Name = "Jose Jacinto", Number = 7890}  
};  
  
// Selecciona alunos que têm José no nome  
var res = from s in stds where s.Name.Contains("Jose") select s.Name;  
  
Console.WriteLine("\n\nALUNOS com o nome 'Jose': \n");  
  
foreach (var s in res)  
    Console.WriteLine("> " + s);  
  
Console.WriteLine("\n2 alunos adicionados");  
stds.Add(new Student { Name = "Carlos Costa", Number = 8901 });  
stds.Add(new Student { Name = "Sa Jose Pinto", Number = 9012 });  
Console.WriteLine("\n\nALUNOS com o nome 'Jose': \n");  
foreach (var s in res)  
    Console.WriteLine("> " + s);
```



```
List<Student> stds = new List<Student> {  
    new Student {Name = "Joao Carlos", Number = 1234},  
    new Student {Name = "Jose Antunes", Number = 2345},  
    new Student {Name = "Joao Silva", Number = 3456},  
    new Student {Name = "Ana Costa", Number = 4567},  
    new Student {Name = "Ivo Tiago", Number = 5678},  
    new Student {Name = "Manuel Jose", Number = 6789},  
    new Student {Name = "Jose Jacinto", Number = 7890}  
};
```

```
// Selecciona alunos que têm José no nome  
var res = from s in stds where s.Name.Contains("Jose") select s.Name;
```



LINQ

Sessão 2 – Projeção de dados

Projeção de dados com tipos anónimos.



```
public class Student
{
    public string Name { get; set; }
    public int Number { get; set; }

    public Student()
    {
    }

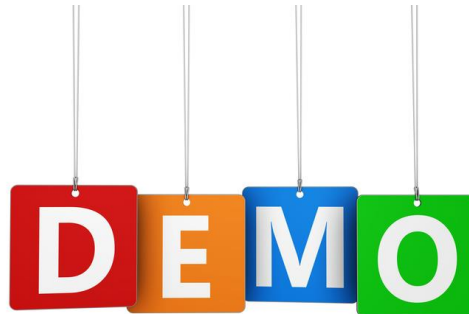
    public Student(string name, int number)
    {
        Name = name;
        Number = number;
    }

    public override string ToString()
    {
        return $"Name: {Name}, Number: {Number}";
    }
}
```




```
List<Student> stds = new List<Student> {  
    new Student {Name = "Joao Carlos", Number = 1234},  
    new Student {Name = "Jose Antunes", Number = 2345},  
    new Student {Name = "Joao Silva", Number = 3456},  
    new Student {Name = "Ana Costa", Number = 4567},  
    new Student {Name = "Ivo Tiago", Number = 5678},  
    new Student {Name = "Manuel Jose", Number = 6789},  
    new Student {Name = "Jose Jacinto", Number = 7890}  
};
```

```
// Selecciona alunos que têm José no nome  
var res = from s in stds where s.Name.Contains("Jose") select s.Name;
```



LINQ

Sessão 3 – LINQ com métodos de extensão

Utilização de LINQ com a sintaxe dos métodos de extensão.

C# - LINQ com a sintaxe de métodos de extensão



```
public class Student
{
    public string Name { get; set; }
    public int Number { get; set; }

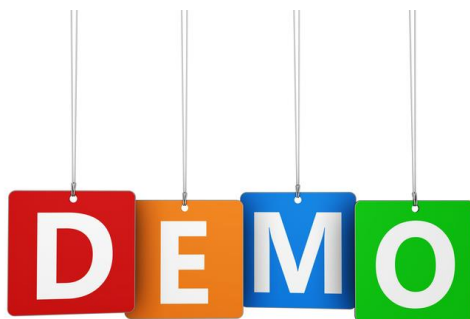
    public Student()
    {
    }

    public Student(string name, int number)
    {
        Name = name;
        Number = number;
    }

    public override string ToString()
    {
        return $"Name: {Name}, Number: {Number}";
    }
}
```

```
List<Student> stds = new List<Student> {
    new Student {Name = "Joao Carlos", Number = 1234},
    new Student {Name = "Jose Antunes", Number = 2345},
    new Student {Name = "Joao Silva", Number = 3456},
    new Student {Name = "Ana Costa", Number = 4567},
    new Student {Name = "Ivo Tiago", Number = 5678},
    new Student {Name = "Manuel Jose", Number = 6789},
    new Student {Name = "Jose Jacinto", Number = 7890}
};

// Selecciona alunos que têm José no nome
var res = from s in stds where s.Name.Contains("Jose") select s.Name;
```



LINQ

Sessão 4 – Criação dos métodos de extensão do LINQ

Tópico Avançado: criar os métodos de extensão tal como é feito no LINQ.



```
public class Person
{
    public string Name { get; set; }
    public int Age { get; set; }

    public override string ToString()
    {
        return $"{Name} ({Age} anos)";
    }
}
```

